



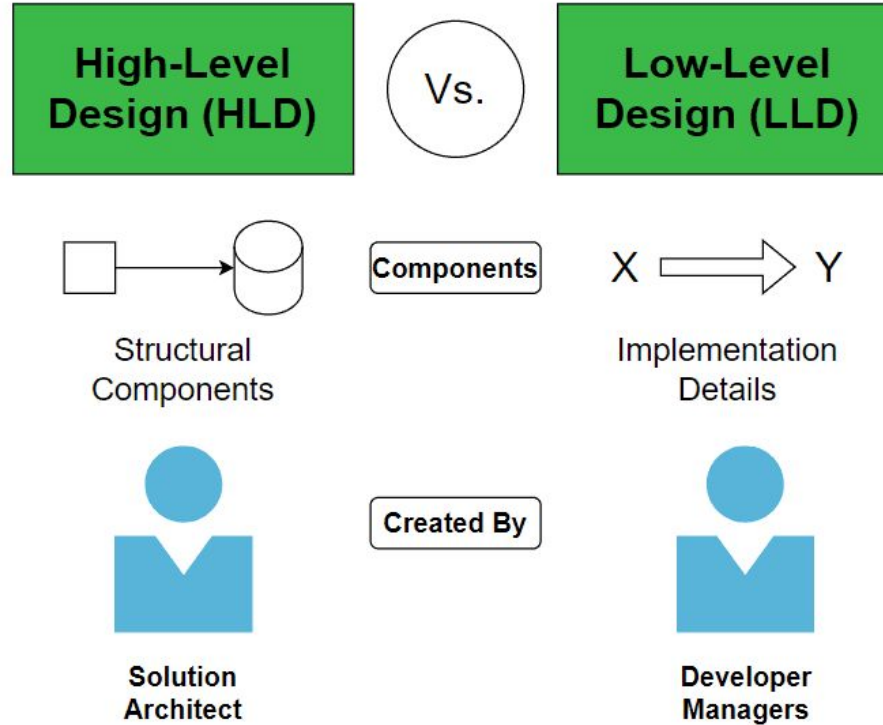
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

Системный дизайн современных приложений

Лекция №4
Компоненты HLD: интеграции



HLD vs LLD





Основные аспекты SD

1. Масштабируемость
2. Производительность
3. Надежность
 - а. Отказоустойчивость
 - б. Доступность
4. Безопасность
5. Адаптивность
6. Управляемость и мониторинг
7. Интеграции

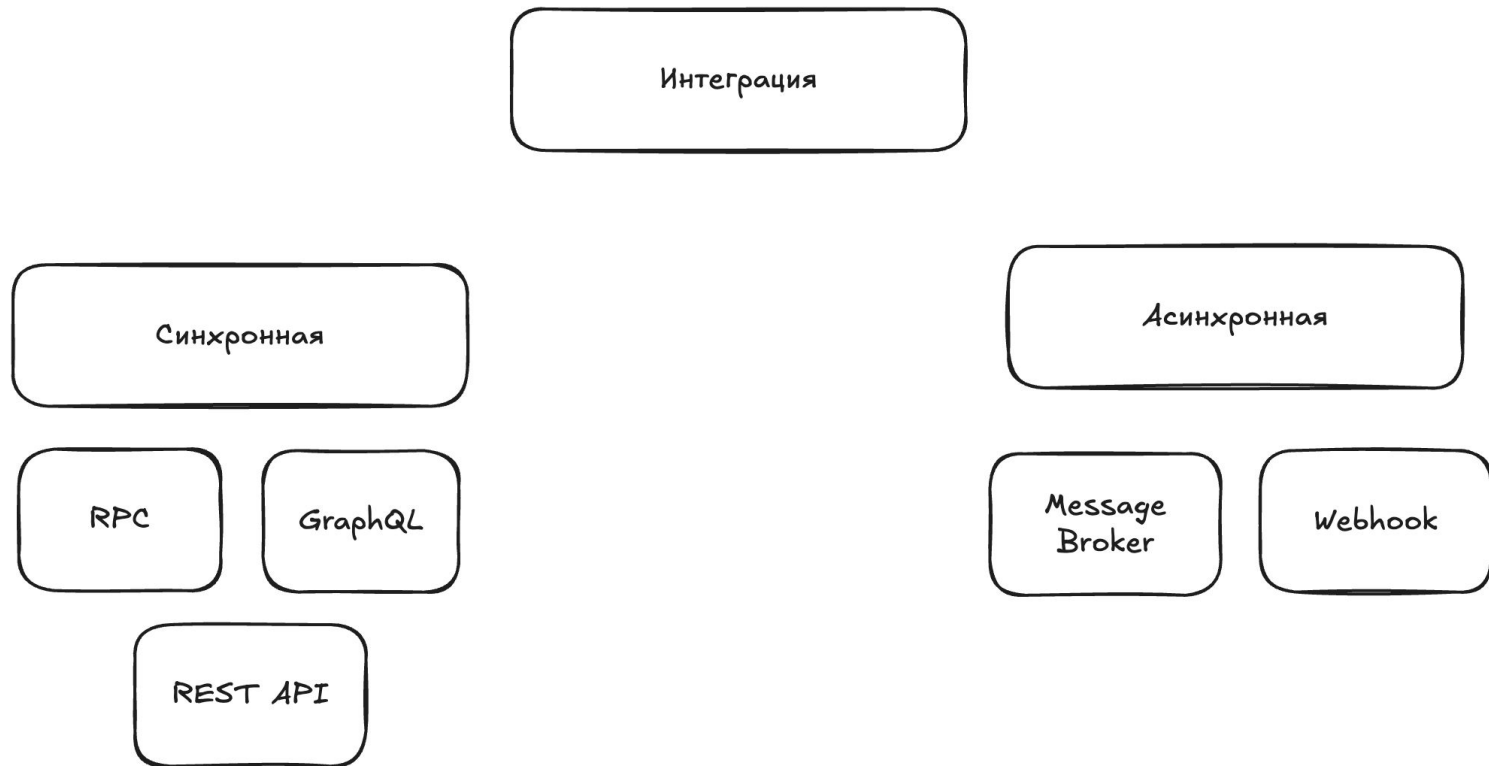


Основные сущности

1. Клиент
2. Сервис
3. Интеграция
4. База данных



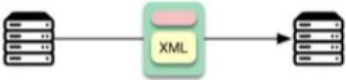


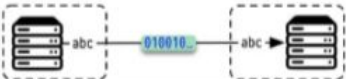
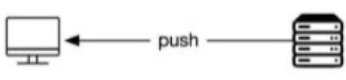
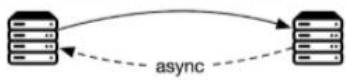
Виды интеграций



Архитектурные стили

API Architecture Styles

ByteByteGo.com

Style	Illustration	Use Cases
SOAP		XML-based for enterprise applications
RESTful		Resource-based for web servers
GraphQL		Query language reduce network load
gRPC		High performance for microservices
WebSocket		Bi-directional for low-latency data exchange
Webhook		Asynchronous for event-driven application



Виды интеграций

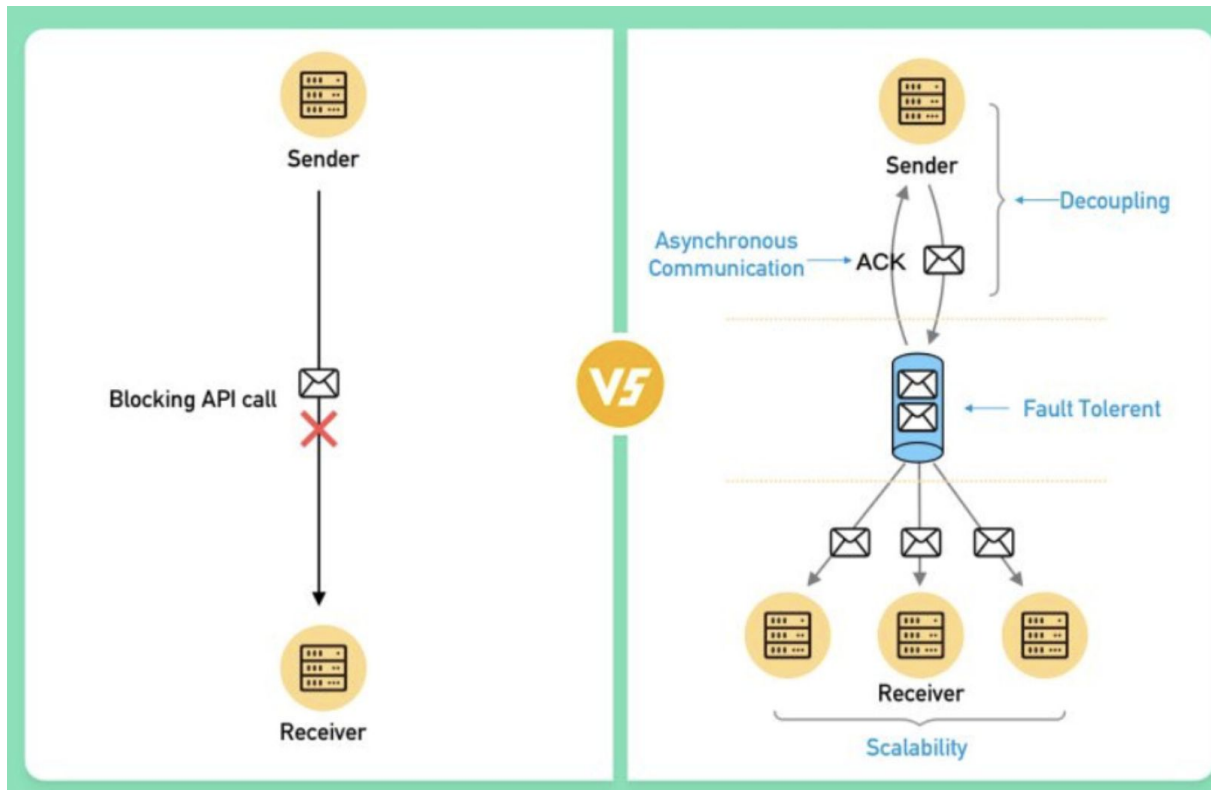
Синхронное

- + Простая реализация
- + Простое тестирование
- + Простая отладка
- Снижение доступности
- Ожидание ответа
- Высокая связанность

Асинхронное

- + Высокая доступность
- + Снижение времени ожидания
- + Слабая связанность
- Сложная реализация
- Сложное тестирование
- Сложная отладка
- Eventual Consistency

Sync vs Async

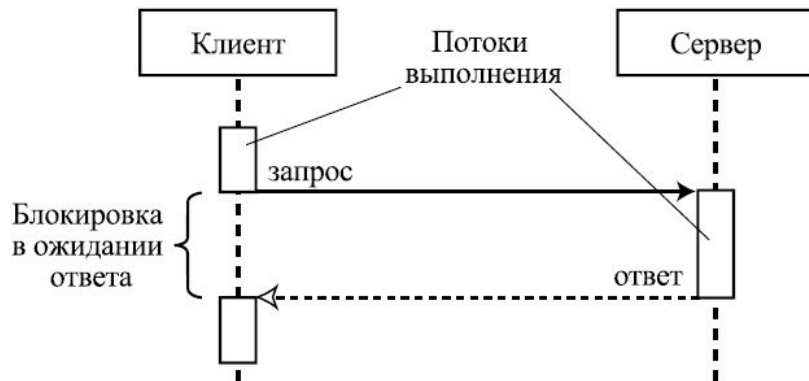




Синхронное взаимодействие

Проблемы

1. Блокировка во время ожидания ответа: память, иные ресурсы.
2. Отказоустойчивость: во время запроса может умереть как вызывающий, так и вызываемый
3. Масштабирование: один экземпляр вызываемого вызывает один экземпляр вызывающего
4. Производительность: таймауты

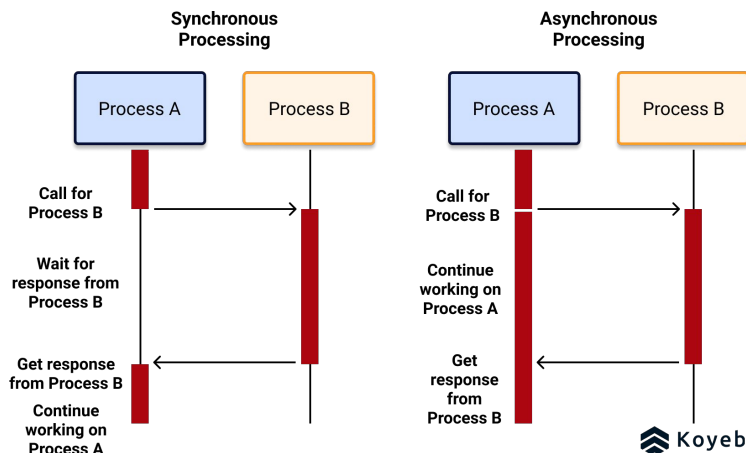




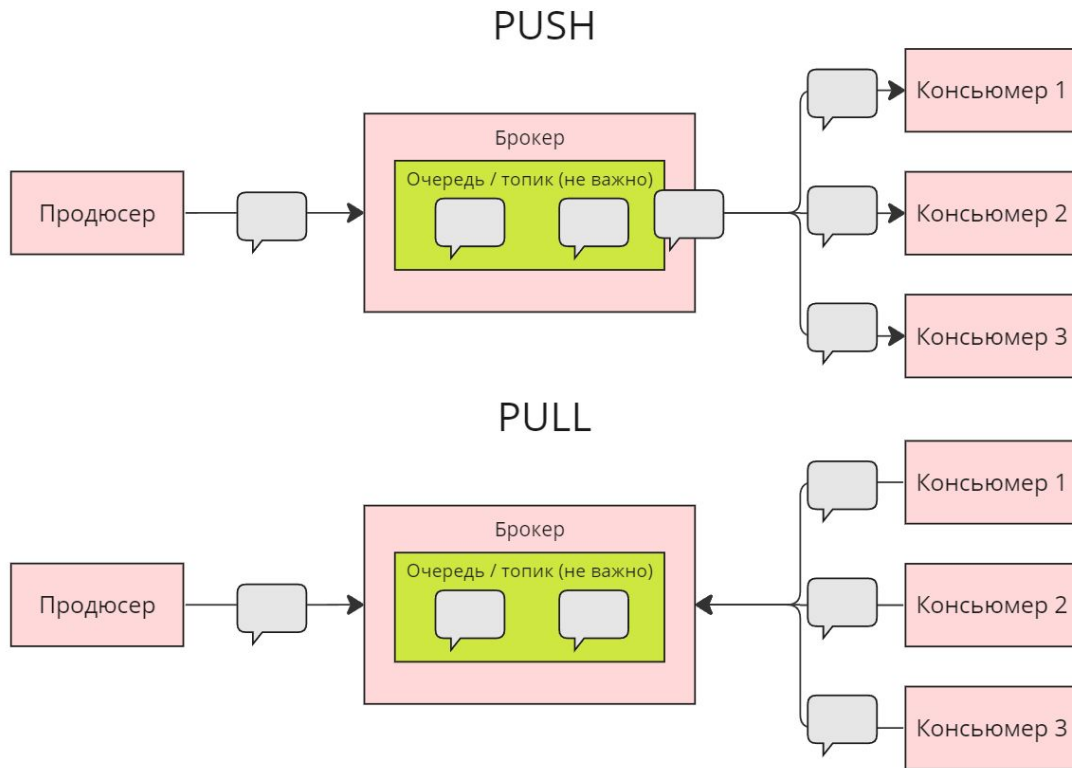
Асинхронное взаимодействие

Проблемы

1. Трудно определить, завершилась ли операция, произошла ли ошибка, или система еще обрабатывает запрос
2. Идемпотентность: один и тот же запрос несколько раз
3. Потерянные сообщения: сообщение может быть отправлено, но не доставлено из-за сетевых сбоев или падения сервиса
4. Порядок сообщений и сложность в отладке



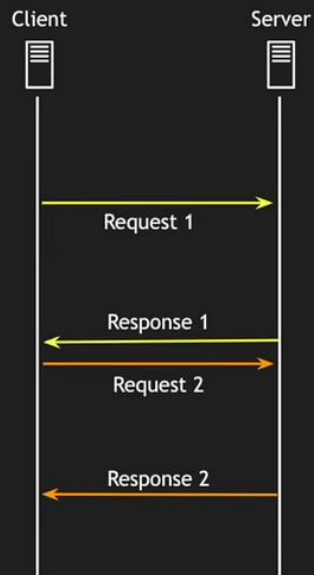
Асинхронное взаимодействие [2]



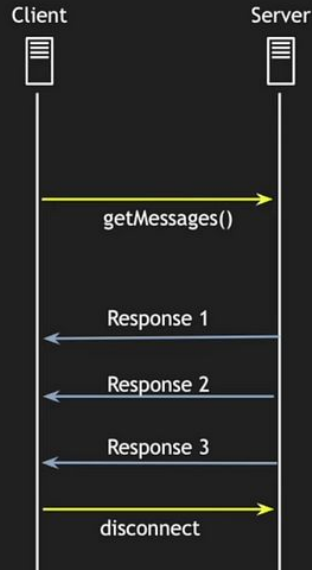


Асинхронное взаимодействие [3]

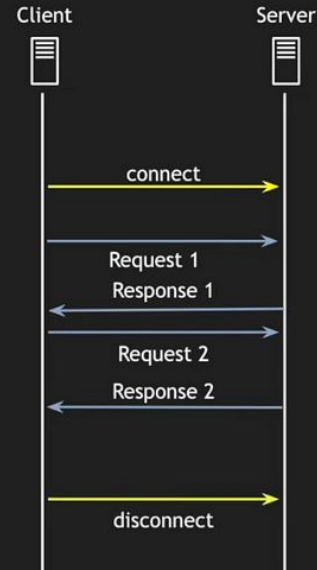
Long Polling



Server Sent Events



WebSockets





Когда выбираем?

1. Получить данные на Frontend для отрисовки корзины товаров
 - а. Синхронное: хотим увидеть товары здесь и сейчас
2. Требуется гарантированная доставка документов
 - а. Асинхронное: можно подумать над ретраями, но лучше Kafka + DLQ
3. Генерация отчета
 - а. Асинхронное: долгая операция, сгенерироваться может и позже
4. Бизнес хочет, чтобы в браузере рендерилось уведомление (работа колокольчика)
 - а. Асинхронное: вебхуки или sse
5. Поточковая обработка данных, например, логирование
 - а. Асинхронное: не требуют мгновенной обработки
6. Перевод денег в Сбере
 - а. Синхронное
7. Бронируем билеты на Aviasales
 - а. Синхронное: нельзя допустить бронь одного места 2 раза

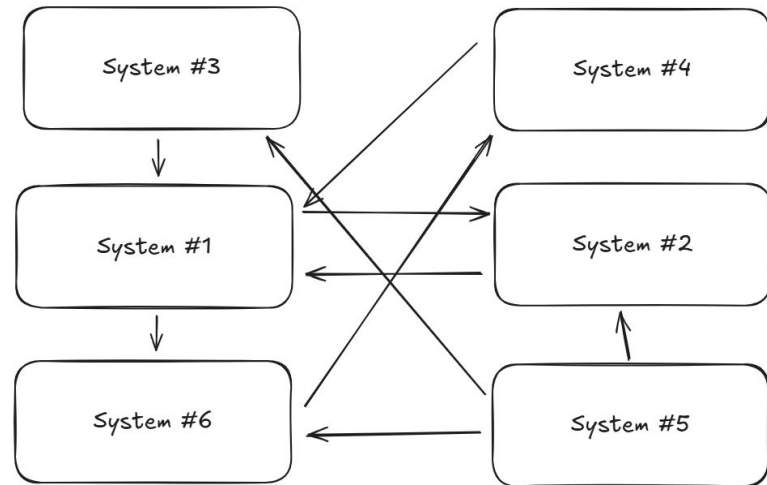
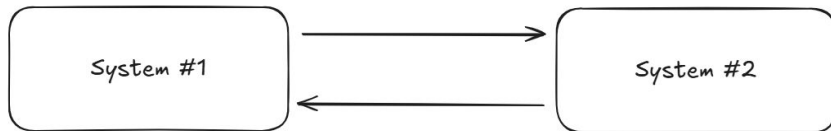


Типы интеграций

1. Точка-точка (P2P)
2. Старые-добрые: через БД, через файловое хранилище
3. Интеграция через ESB (Event-Source Bus)
4. Интеграция через брокер сообщений
5. Интеграция через API-шлюз (API Gateway)

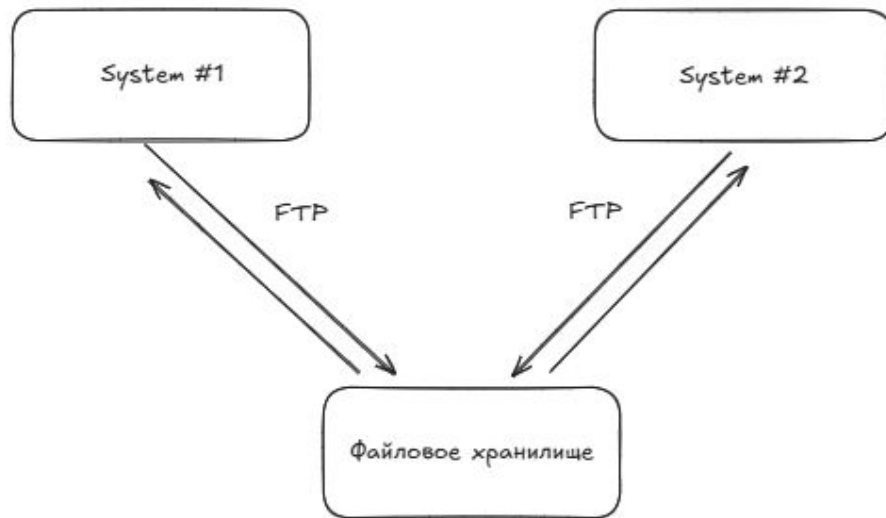
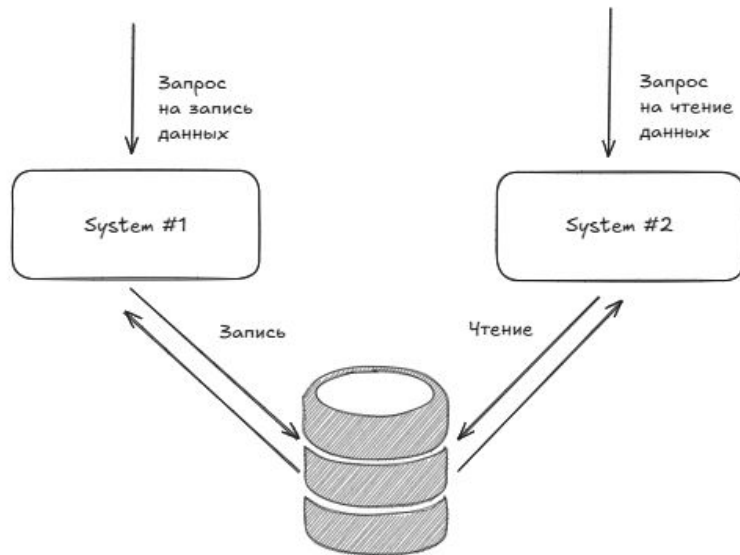
Определение

Прямая интеграция 2-х систем без посредников



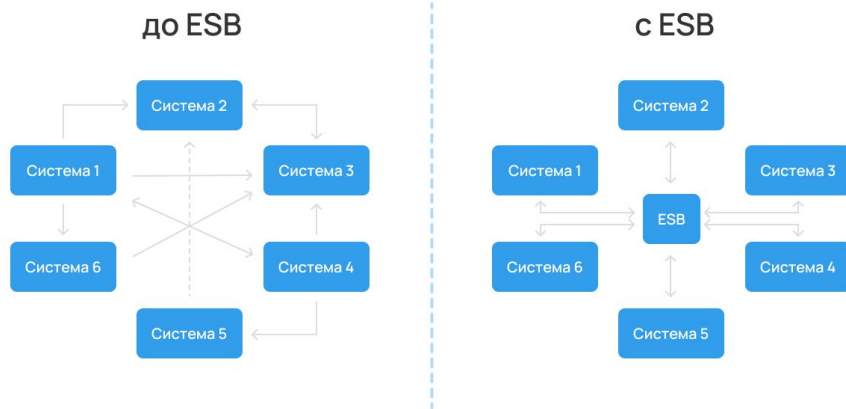
Определение

Интеграция 2-х систем с посредником в виде базы данных/файлового хранилища



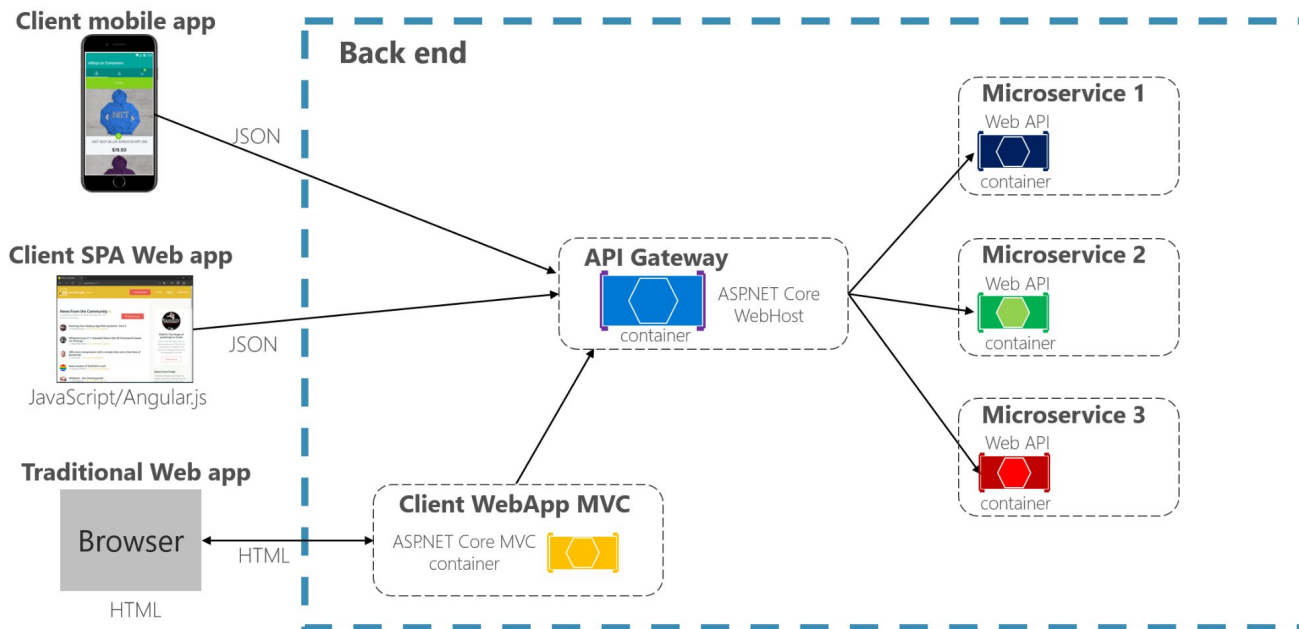
Определение

Интеграция систем с посредником в виде централизованной шины сервисов

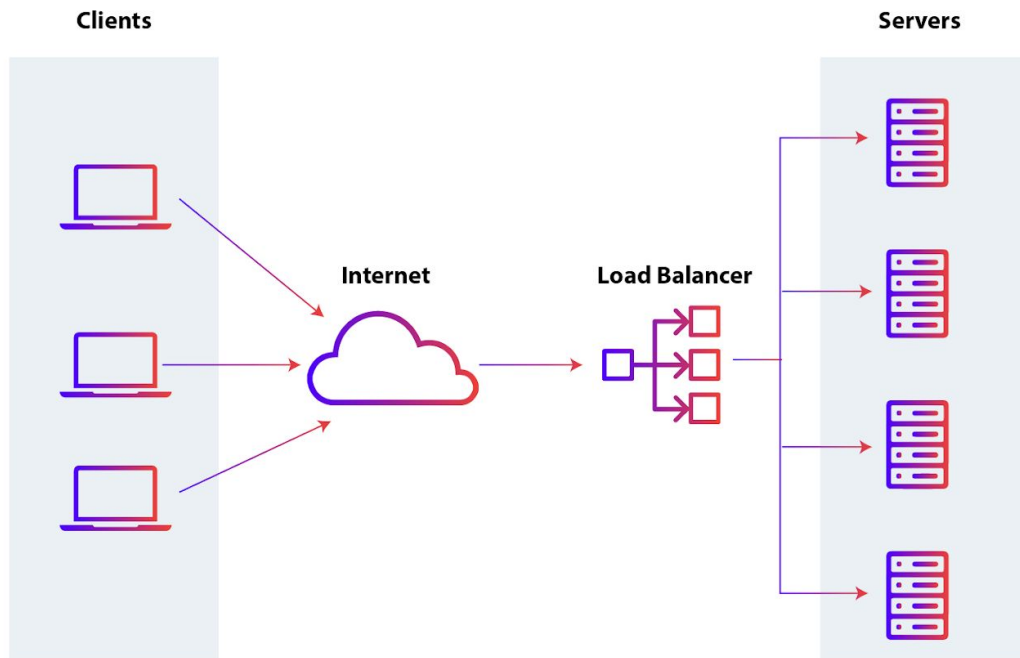


API Gateway

Using a single custom **API Gateway service**



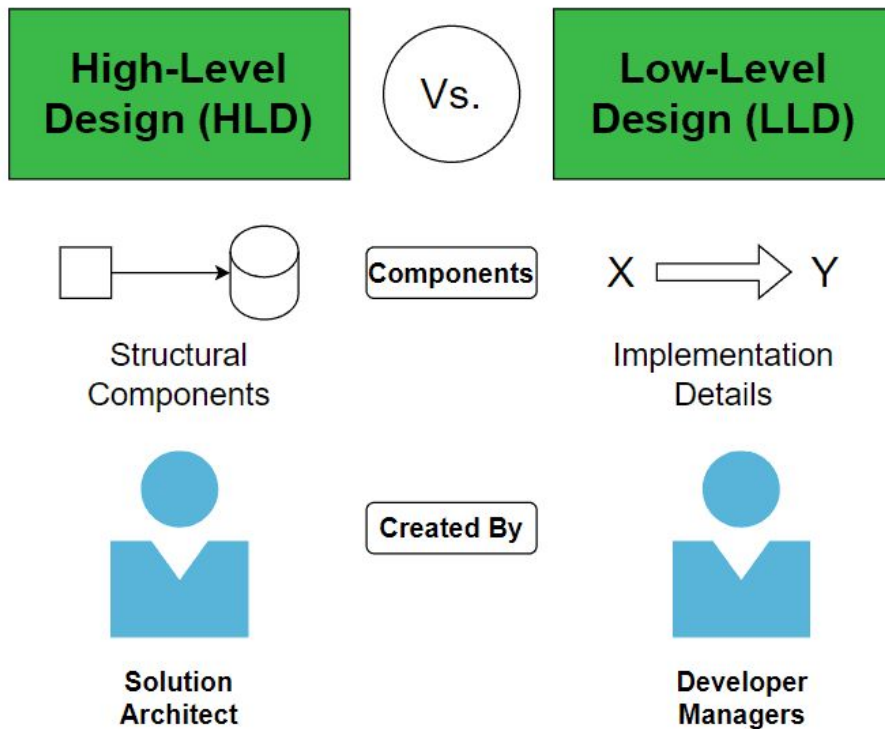
Балансировщик нагрузки





НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

HLD vs LLD





Основные аспекты SD

1. Масштабируемость
2. Производительность
3. Надежность
 - а. Отказоустойчивость
 - б. Доступность
4. Безопасность
5. Адаптивность
6. Управляемость и мониторинг
7. Интеграции

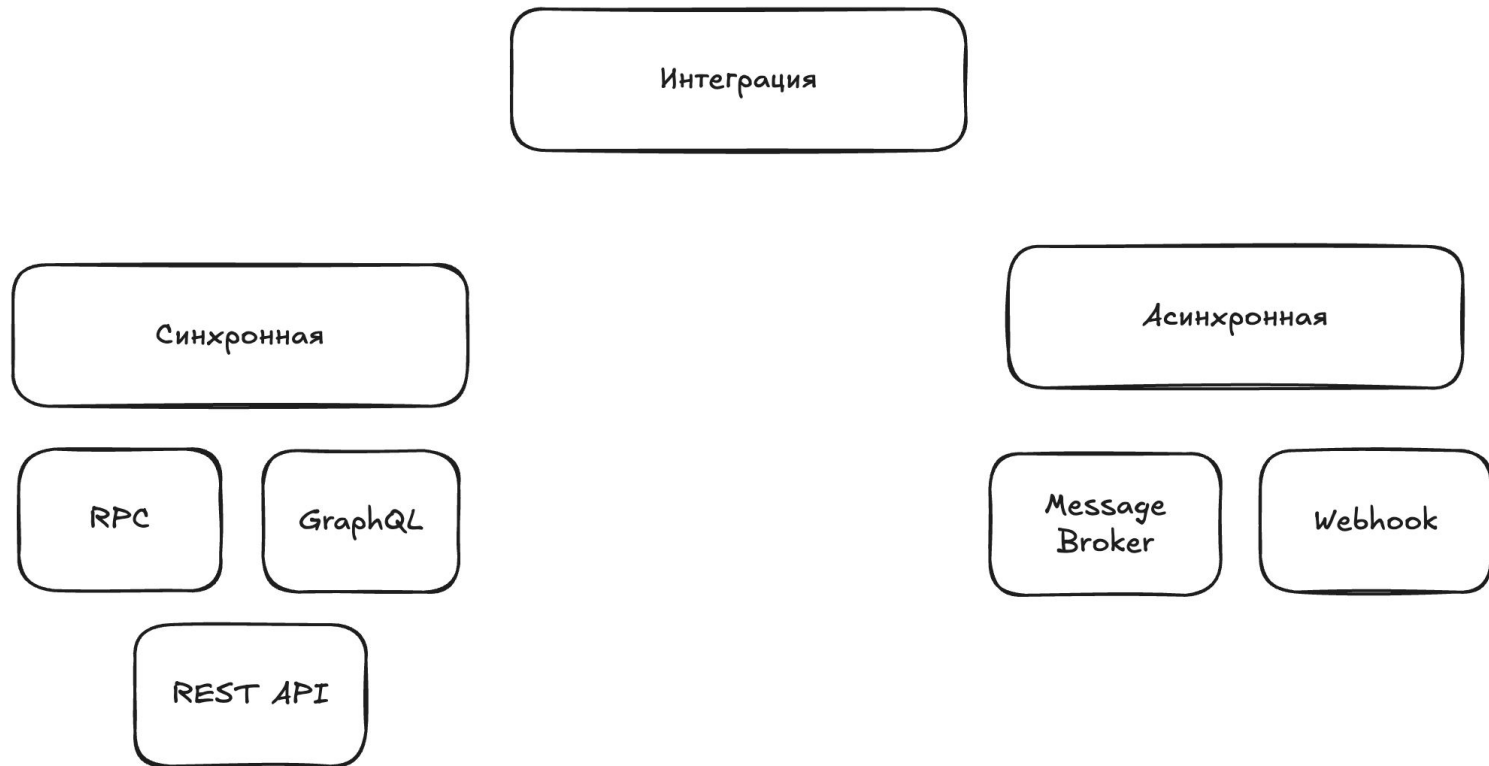


Основные сущности

1. Клиент
2. Сервис
3. Интеграция
4. База данных



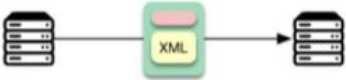


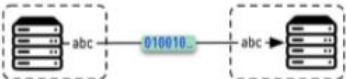
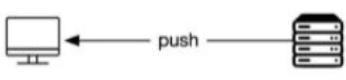
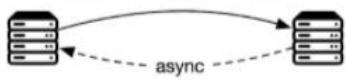
Виды интеграций



Архитектурные стили

API Architecture Styles

ByteByteGo.com

Style	Illustration	Use Cases
SOAP		XML-based for enterprise applications
RESTful		Resource-based for web servers
GraphQL		Query language reduce network load
gRPC		High performance for microservices
WebSocket		Bi-directional for low-latency data exchange
Webhook		Asynchronous for event-driven application

Виды интеграций

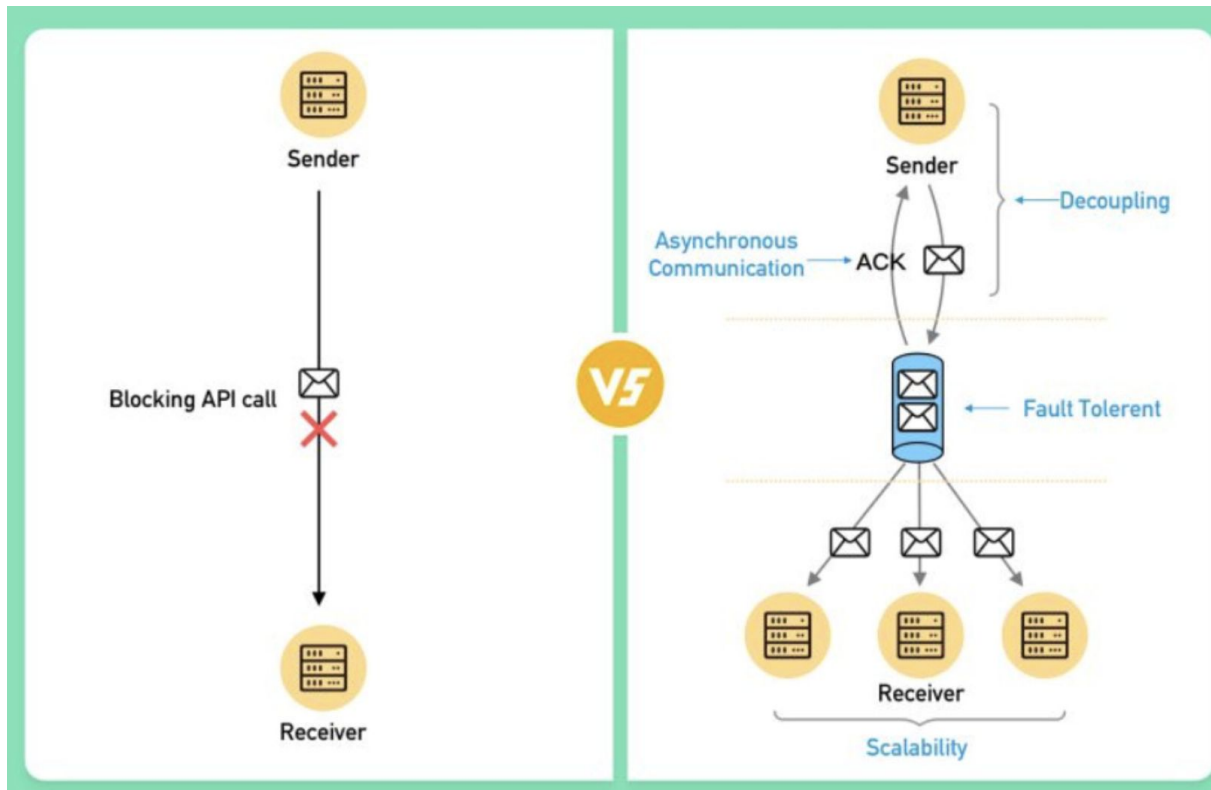
Синхронное

- + Простая реализация
- + Простое тестирование
- + Простая отладка
- Снижение доступности
- Ожидание ответа
- Высокая связанность

Асинхронное

- + Высокая доступность
- + Снижение времени ожидания
- + Слабая связанность
- Сложная реализация
- Сложное тестирование
- Сложная отладка
- Eventual Consistency

Sync vs Async

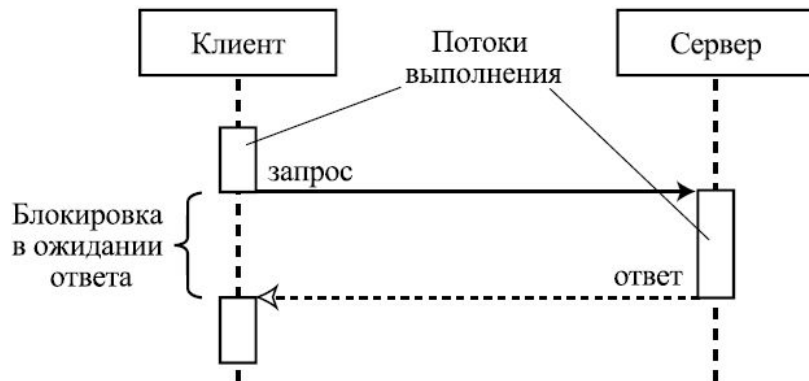




Синхронное взаимодействие

Проблемы

- 1) Блокировка во время ожидания ответа: память, иные ресурсы.
- 2) Отказоустойчивость: во время запроса может умереть как вызывающий, так и вызываемый
- 3) Масштабирование: один экземпляр вызываемого вызывает один экземпляр вызывающего
- 4) Производительность: таймауты

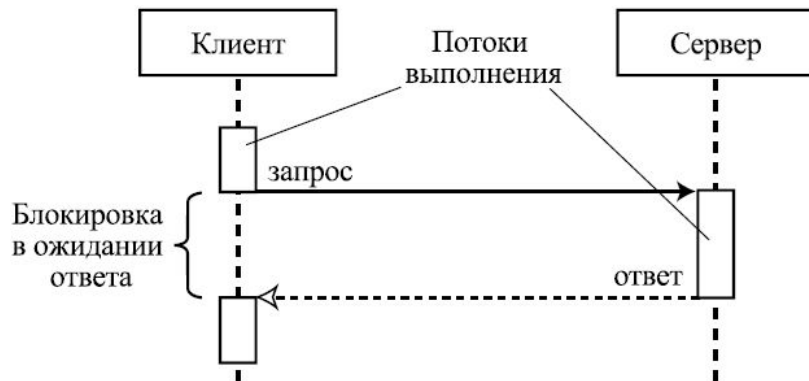




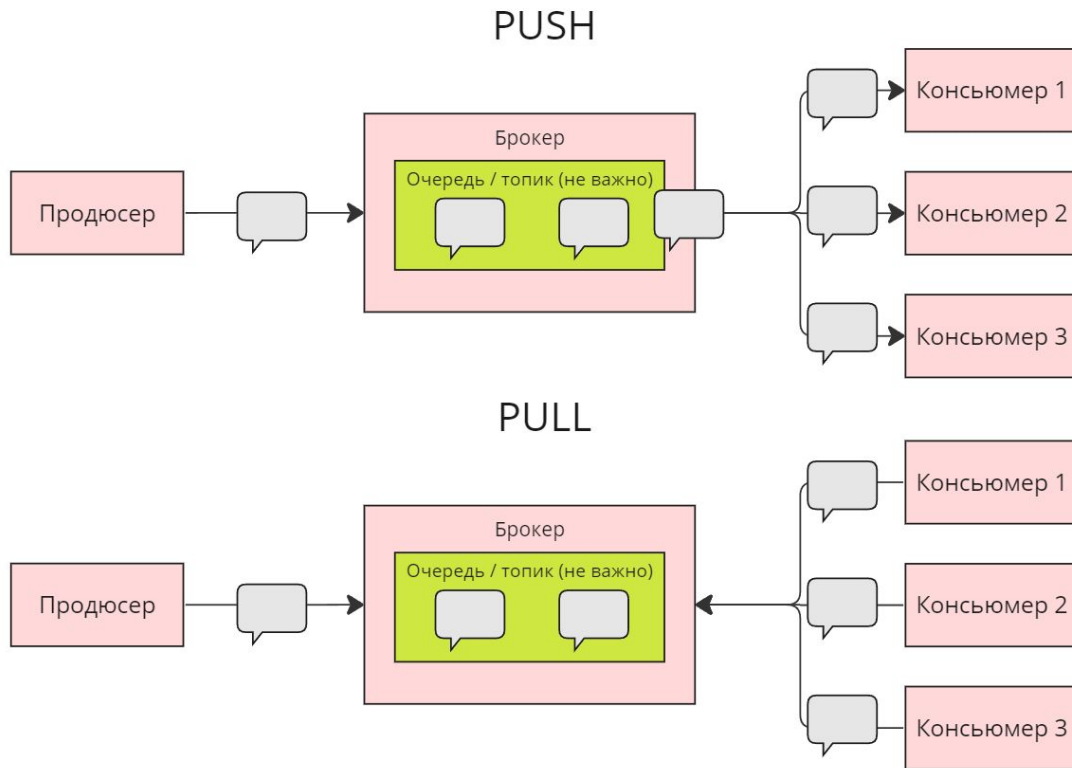
Асинхронное взаимодействие

Проблемы

- 1) Блокировка во время ожидания ответа: память, иные ресурсы.
- 2) Отказоустойчивость: во время запроса может умереть как вызывающий, так и вызываемый
- 3) Масштабирование: один экземпляр вызываемого вызывает один экземпляр вызывающего
- 4) Производительность: таймауты



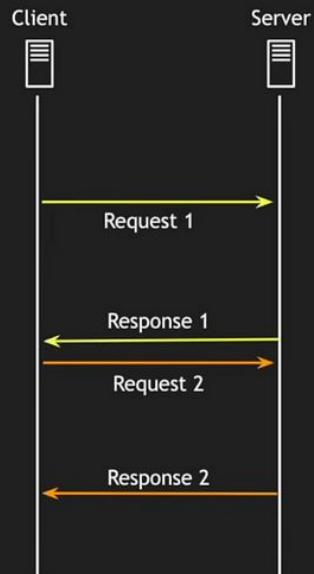
Асинхронное взаимодействие [2]



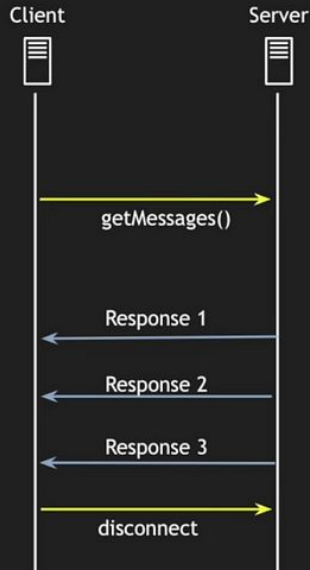


Асинхронное взаимодействие [3]

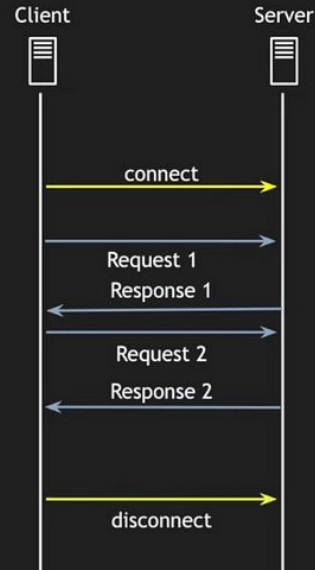
Long Polling



Server Sent Events



WebSockets





Когда выбираем?

1. Получить данные на Frontend для отрисовки корзины товаров
 - а. Синхронное: хотим увидеть товары здесь и сейчас
2. Требуется гарантированная доставка документов
 - а. Асинхронное: можно подумать над ретраями, но лучше Kafka + DLQ
3. Генерация отчета
 - а. Асинхронное: долгая операция, сгенерироваться может и позже
4. Бизнес хочет, чтобы в браузере рендерилось уведомление (работа колокольчика)
 - а. Асинхронное: вебхуки или sse
5. Поточковая обработка данных, например, логирование
 - а. Асинхронное: не требуют мгновенной обработки
6. Перевод денег в Сбере
 - а. Синхронное
7. Бронируем билеты на Aviasales
 - а. Синхронное: нельзя допустить бронь одного места 2 раза

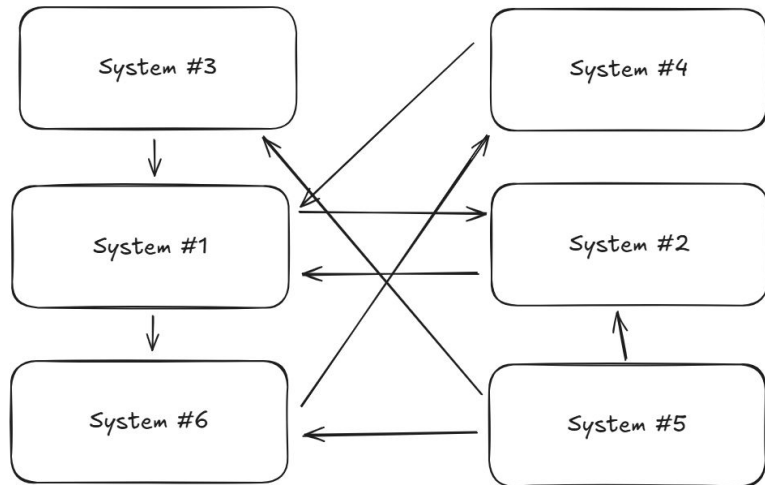
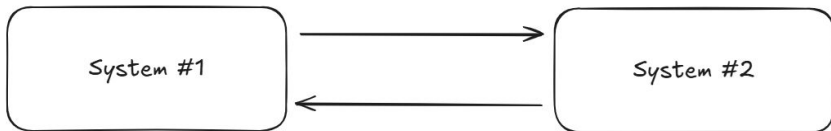


Типы интеграций

1. Точка-точка (P2P)
2. Старые-добрые: через БД, через файловое хранилище
3. Интеграция через ESB (Event-Source Bus)
4. Интеграция через брокер сообщений
5. Интеграция через API-шлюз (API Gateway)

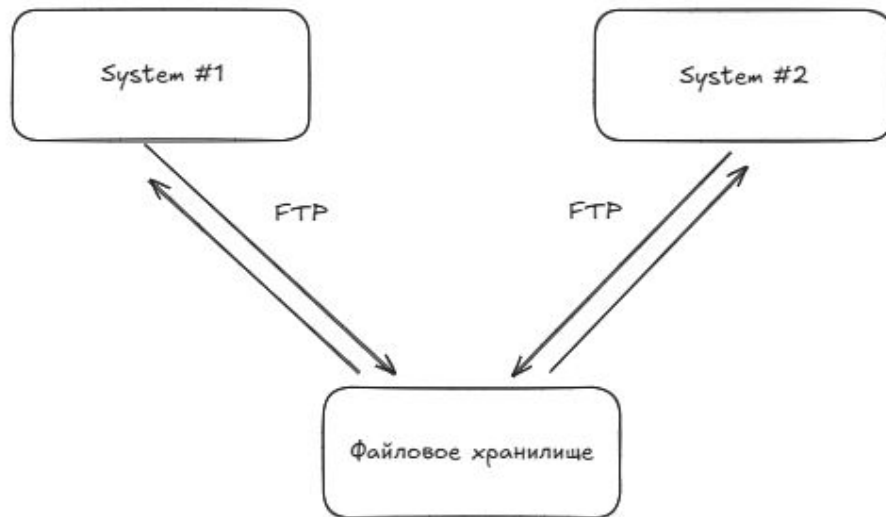
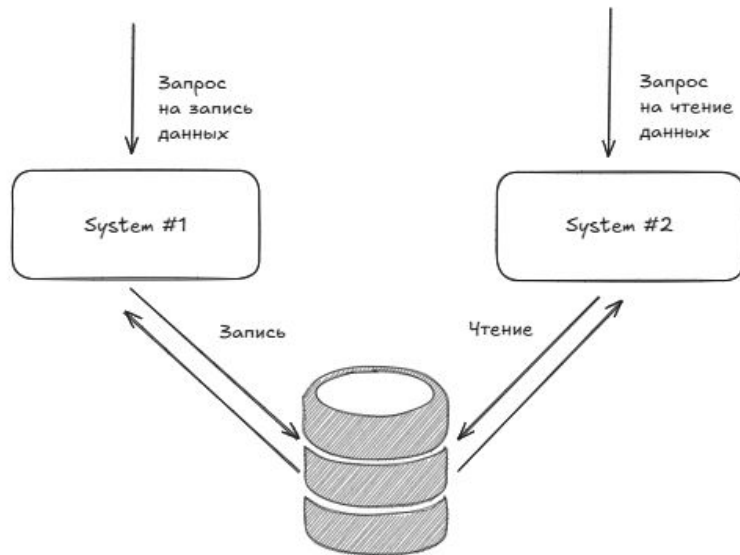
Определение

Прямая интеграция 2-х систем без посредников



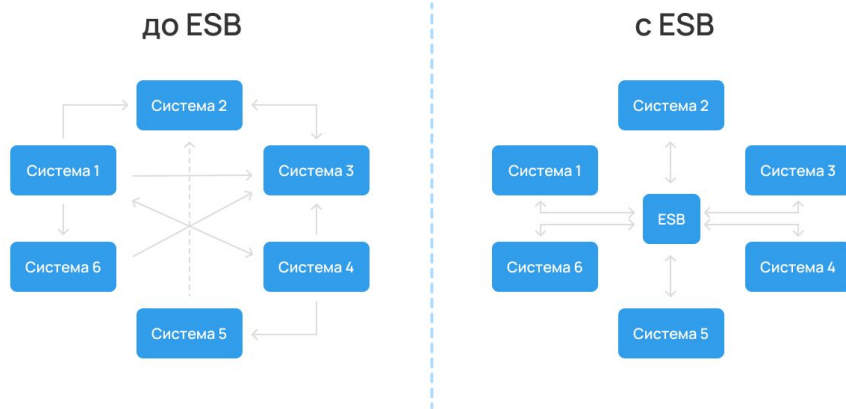
Определение

Интеграция 2-х систем с посредником в виде базы данных/файлового хранилища



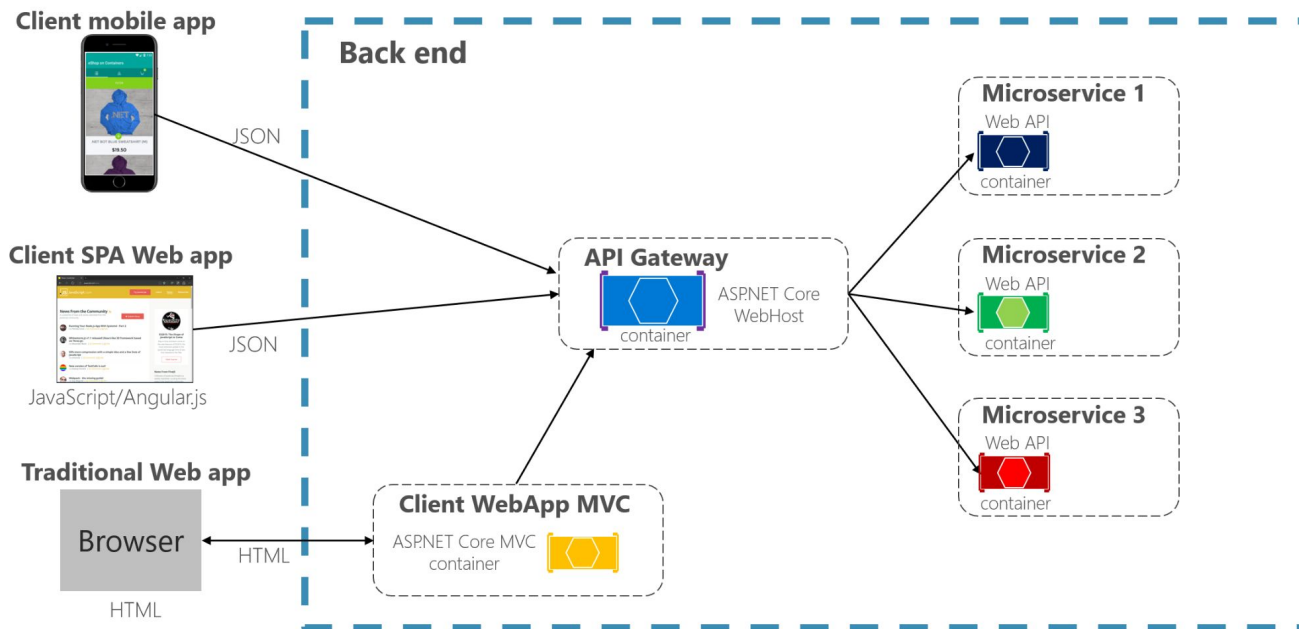
Определение

Интеграция систем с посредником в виде централизованной шины сервисов

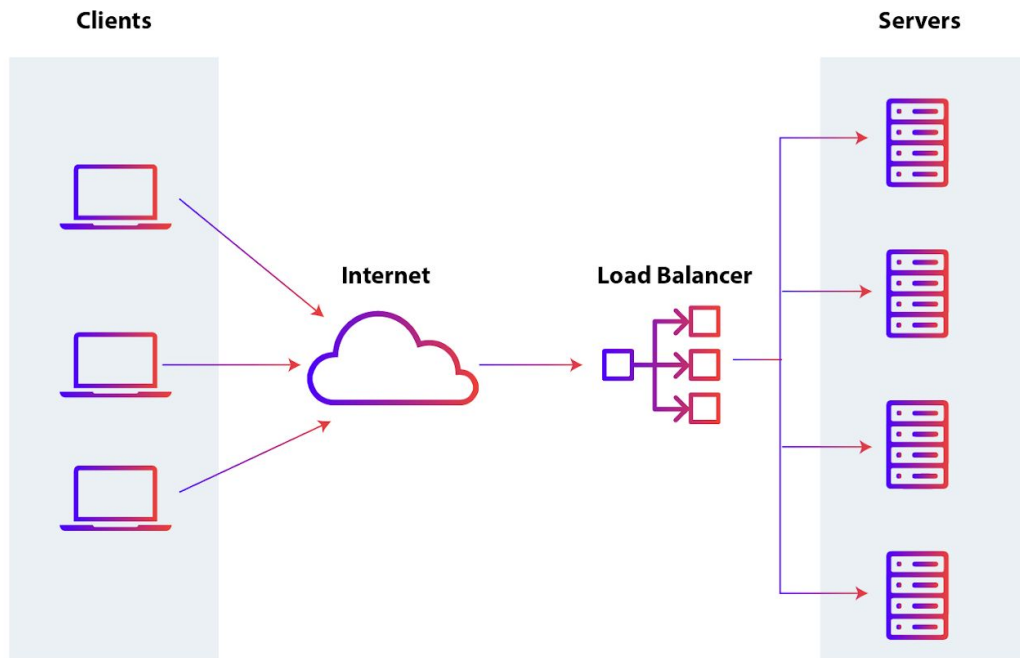


API Gateway

Using a single custom **API Gateway service**



Балансировщик нагрузки





НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ